

IMPROVEMENTS FOR B.E.M. IMPLEMENTATION IN MICROS

E.ALARCON , C.GARCIA-SUAREZ , A.REVERTER

E.T.S.ING.INDUSTRIALES . UNIVERSIDAD POLITECNICA
MADRID . SPAIN

SUMMARY

This paper presents a computer program developed to run in a micro I.B.M.-P.C. , which incorporates some features in order to optimize the number of operations needed to compute the solution of plane potential problems governed by Laplace's equation by using the Boundary Integral Equation Method (B.I.E.M.) . Also incorporated is a routine to plot isolines inside the domain under study .

1. INTRODUCTION

After some years of intensive research B.I.E.M. is being accepted as a useful tool in many branches of engineering . Among them , the solution of plane potential problems continues to be an everyday question and B.I.E.M. is specially well suited to solve it . As is well known , one of the main advantages of the procedure is the need to discretize only the boundary , in order to obtain the problem unknowns . The basis is an integral relationship , or 'representation formula'

$$c \phi + \int_{\partial\Omega} \phi q^* ds = \int_{\partial\Omega} q \phi^* ds \quad \dots (1)$$

that can be repeated as many times as needed using different poles to obtain the 'fundamental solution'

$$\phi^*, q^*$$

In this way it is possible to establish N equations to determine the parameters $a_1, a_2, \dots, b_1, b_2, \dots$ in

$$\phi = a_j \phi_j$$

... (2)

$$q = b_j q_j$$

to interpolate both the potential ϕ and the fluxes q . Of course it is necessary also to know the boundary conditions in order to reduce the 2 N parameters a_j, b_j to N unknowns.

2. PROGRAM PROCEDURE

The program follows the steps indicated in fig.1. The degree of interpolation used in it corresponds to a classical linear interpolation both in unknowns and in geometry. In order to reduce the number of operations to a minimum every piece of information that can be used repeatedly (length, normals, integration points, etc) has been stored but also the ordering of operations has been composed in such a form to sequentially reuse the same space of core storage. In this way a simple and efficient, as well as economic, procedure has been implemented allowing the solution of usual problems with speed and accuracy.

In addition the analysis of boundary conditions has been carefully considered in the light of previous experience. In this program we incorporate the so-called 'corner element' to take account of sharp Dirichlet-type corners, where apparently the number of unknowns is 2 and the collocation equation is only one. On the contrary, due to its sharpness that kind of corner is totally determined because it is possible to obtain, by derivation along the two corner sides, the local gradient components and, then, by projection on the respective normals the corresponding fluxes around the corner. In this way it is possible to reduce the

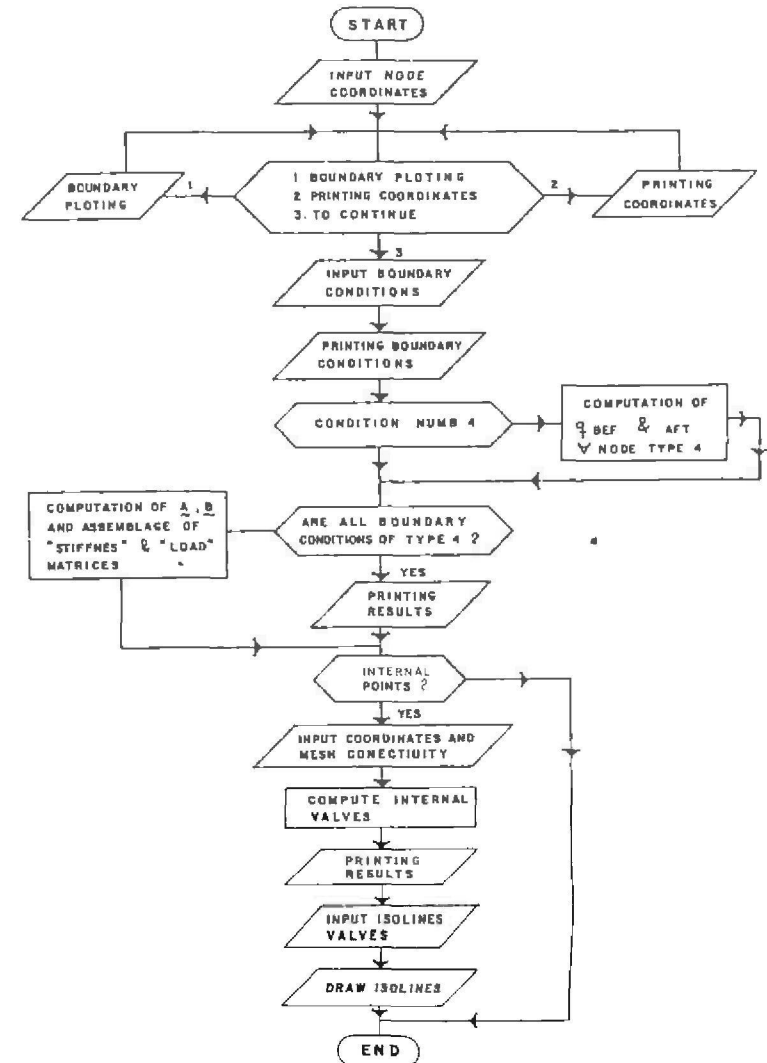


FIGURE 1.

number of equations to be solved , being zero in the limit if all the nodes are of that type. Figure 1 and the accompanying listing are selfexplaining.

3. REFERENCES

1. ZIENKIEWICZ , O.C., The finite element method (3rd. ed) Mc.Graw

2. ALARCON , E.; MARTIN , A.; PARIS , F., Boundary elements in potential and elasticity theory. Computers and Structures .10 . 1979

3. ALARCON , E. Letter to the editor . International Journal of Numerical Meth.in Eng. 19 .7 .p.1105.1983

```

50000 '
50010 ' This subroutine computes the values of the fuction at inside points
50020 '
50030 CLS:PRINT:PRINT :PRINT "Now you can compute the function at any point insi
de the domain."
50040 PRINT "If desired You can draw isolines so."
50050 PRINT:PRINT:PRINT "Enter Y if you want any of those options, otherwise pre
ss any key."
50060 A$=INKEY$:IF A$="" THEN GOTO 50060
50070 IF (A$(">")"Y" AND A$(">")"y") THEN END
50080 '
50090 '
50100 ' The control is transferred to a second segment being at the disk
50110 ' If this program is wanted to work all together just copy downwars
50120 ' the program located at "inside in this disk"
50130 '
50140 '
50150 OPEN "paso" FOR OUTPUT AS #1
50160 PRINT #1,NBE
50170 FOR I=1 TO NBE
50180 PRINT #1,COND(I,1),COND(I,2),COND(I,3),COND(I,4)
50190 NEXT I
50191 FOR I=1 TO NBE
50192 PRINT #1,X(I),Y(I)
50193 NEXT I
50200 FOR TIP=1 TO 2
50210 FOR NP1=1 TO 4
50220 PRINT #1,M1(TIP,NP1),M2(TIP,NP1),W(TIP,NP1)
50230 NEXT NP1
50240 NEXT TIP
50241 CLOSE #1
50250 ' CHAIN "inside"

```

```

1 ' PROGRAM CABEM
2 ' October, 1983
3 ' Dept. Estructuras
4 ' Escuela Técnica Superior de Ingenieros Industriales
5 ' C/José Gutierrez Abascal 2, Madrid 6, Spain
6 '
10 SCREEN 0
20 OPTION BASE 1
30 KEY OFF
40 PRINT CHR$(12):COLOR 2
50 FOR I=1 TO 5
60 PRINT " "
70 NEXT I
80 PRINT " This program solves plane potential problems inside simply "
81 PRINT " connected domains when described by Laplace's equation. "
90 PRINT " The program gives the values of the unknown "
100 PRINT " function and the flux along the boundary. The function values"
110 PRINT " can be computed at any interior point previously indicated."
120 PRINT " The lines of constant values for the function can be drawn if"
130 PRINT " desired."

140 FOR I=1 TO 2
150 PRINT " "
160 NEXT I
170 PRINT "Press any key to continue (Is printer ready?)"
180 A$=INKEY$:IF A$="" THEN 180
190 '
200 '
210 ' The geometry of the mesh is read and/or generated
220 '
230 PRINT CHR$(12)
240 INPUT "number of boundary elements = ";NBE
250 '
260 "### I(i) and Y(i) are the coordinates of the boundary nodes
270 '
280 DIM X(NBE),Y(NBE)
290 PRINT CHR$(12)
300 FOR I=1 TO 4:PRINT "":NEXT I
310 PRINT " Now you are required to give the boundary geometry. "
320 PRINT " The program has been prepared to allow partial "
330 PRINT " automatic generation."
340 PRINT " When the coordinantes of a node is not explicitly given."
350 PRINT " linear interpolation is assumed."
360 FOR I=1 TO 2:PRINT "":NEXT I:PRINT " Press any key to continue"
370 A$=INKEY$:IF A$="" THEN 370
380 PRINT CHR$(12)
381 PRINT "Do you want this data to be read from any file ? (Y o N)
382 A$=INKEY$:IF A$="" THEN GOTO 382
383 IF (A$(">")"Y" AND A$(">")"y") THEN GOTO 390
384 CLS:INPUT "enter file name";B$
385 OPEN B$ FOR INPUT AS#1
386 FOR NOD0=1 TO NBE

```

```

387 INPUT #1, X(NDD0),Y(NDD0)
388 NEXT NDD0
389 CLOSE #1:GOTO 560
390 CLS:PRINT " Mode Number,X coordinate,Y coordinate (enter in that order
)"
400 PRINT:PRINT:PRINT
410 K=0
420 FOR I=1 TO NBE
430 IF (X=NBE) THEN GOTO 570
440 INPUT " ";L,X(L),Y(L)
450 IF (L=(K+1)) THEN K=K+1:GOTO 550
460 INCRE=L-K
470 AR=(X(L)-X(K))/INCRE
480 AZ=(Y(L)-Y(K))/INCRE
490 IN =INCRE *1
500 FOR M=1 TO IN
510 X(K+M)=X(K)+AR*M
520 Y(K+M)=Y(K)+AZ*M
530 NEXT M
540 K=K+INCRE
550 NEXT I
560 ' The boundary geometry has been already read
570 PRINT CHR$(12)
580 PRINT "Select desired code : "
590 PRINT*:PRINT*
600 PRINT " 1)To draw the boundary"
610 PRINT " 2)To print the coordinates"
620 PRINT " 3)To go on with the data input process"
640 INPUT "....";CODE
650 ON CODE GOSUB 2230,1410,1730
655 IF (COND4<>0) THEN GOSUB 3000
656 IF COND4=NBE THEN GOTO 666
660 IF FLAG=1 THEN GOSUB 680 ELSE GOTO 570
661 'call solution subroutine
662 GOSUB 11000
663 'call interpretation subroutine
664 GOSUB 15000
665 'call printing solution subroutine
666 GOSUB 20000
667 'call inside points subroutine
668 GOSUB 50000
670 END
680 '
690 ' computation of influence matrices via BEN
700 '
701 CLS:PRINT "The system is being assembled"
702 NEC=NBE-COND4
705 DIM COEF(NEC,NEC+1),ADIAG(NEC)
706 JCOND4=0
707 ICOND4=0
710 DIM W(2,4),EPSI(2,4),M1(2,4),M2(2,4)
810 DATA -.8611363115,-.3399810435,.3399810435,.8611363115,-0.57735026,0.5773502
6,0,0

```

```

820 DATA 0.3478548451,0.6521451548,0.6521451548,0.3478548451,1,1,0,0
825 FOR ES=1 TO 2
830 FOR I=1 TO 4
840 READ EPSI(ES,I)
850 NEXT I
855 NEXT ES
856 FOR ES=1 TO 2
860 FOR I=1 TO 4
870 READ W(ES,I)
880 NEXT I
885 NEXT ES
886 FOR ES=1 TO 2
890 FOR I=1 TO 4
900 M1(ES,I)=(1-EPSI(ES,I))/2
910 M2(ES,I)=(1+EPSI(ES,I))/2
920 NEXT I
930 NEXT ES
935 NFI=0
940 FOR EL=1 TO NBE
950 JEL=EL+1
960 IF (EL=NBE) THEN JEL=1
962 IF EL=1 AND COND(EL,1)<>4 THEN NFI=1
965 IF EL>1 THEN NFI1=NFI2
967 IF COND(JEL,1)<>4 THEN NFI2=NFI1+1 ELSE NFI2=NFI1
968 IF JEL=1 THEN NFI2=1
970 X1=X(EL)
980 Y1=Y(EL)
990 X2=X(JEL)
1000 Y2=Y(JEL)
1010 DX=X(JEL)-X(EL)
1020 DY=Y(JEL)-Y(EL)
1030 LONG=SQR(DX*DX+DY*DY)
1040 UM1=DY/LONG
1045 UM2=-DX/LONG
1046 B11=(3/4-.5*LOG(LONG))*LONG
1047 B22=(1/4-.5*LOG(LONG))*LONG
1050 FOR NDD0=1 TO NBE
1052 IF COND(NDD0,1)=4 THEN GOTO 1355
1055 A1=0:A2=0
1060 IF (NDD0=EL) THEN B1=B11:B2=B22:GOTO 1305
1070 IF (NDD0=JEL) THEN B2=B11:B1=B22:GOTO 1305
1090 XP=X(NDD0)
1100 YP=Y(NDD0)
1110 B1=0 :B2=0
1120 DX=X1-XP
1130 DY=Y1-YP
1140 R=SQR(DX*DX+DY*DY)
1150 UR1=DX
1160 UR2=DY
1170 DIS=UR1*UM1+UR2*UM2
1172 ' ***criteria for computing npi should be introduced
1173 NPI=4:TIP=1

```

```

1180      FOR K=1 TO NP1
1181          XLOC=N1(TIP,K)*X1+N2(TIP,K)*X2
1182          YLOC=N1(TIP,K)*Y1+N2(TIP,K)*Y2
1190          R=SQR((XLOC-XP)^2+(YLOC-YP)^2)
1200          G=LOG(R)*W(TIP,K)
1210          B1=B1-N1(TIP,K)*G
1220          B2=B2-N2(TIP,K)*G
1230          H=DIS/R^2*W(TIP,K)
1240          A1=A1-H*N1(TIP,K)
1250          A2=A2-H*N2(TIP,K)
1260      NEXT K
1270      A1=A1*LONG/2
1280      A2=A2*LONG/2
1290      B1=B1*LONG/2
1300      B2=B2*LONG/2
1305      NCO=NODO-ICOND4
1310      ADIAG(NCO)=ADIAG(NCO)+A1+A2
1320      NNMN=EL:IND=1
1330      GOSUB 9000
1340      NNMN=JEL:IND=2
1350      GOSUB 9000
1352      GOTO 1360
1355      ICOND4=ICOND4+1
1360  NEXT NODO
1365  ICOND4=0
1369  NEXT EL
1370  ICOND4=0
1371  ' assembly of elements in the main diagonal
1372  FOR I=1 TO NBE
1373      CODE=COND(I,1)
1374      IF (CODE=4) THEN ICOND4=ICOND4+1: GOTO 1379
1375      NCO=I-ICOND4:A=-ADIAG(NCO)
1376      IF (CODE=2 OR CODE=3) THEN COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*COND(I,2)
1377      IF (CODE=5) THEN COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*COND(I,2)
1378      IF (CODE=1) THEN COEF(NCO,NCO)=COEF(NCO,NCO)+A
1379  NEXT I
1400  RETURN
1405  '
1406  '
1410  '   Printing of nodes coordinates"
1420  PRINT CHR$(12)
1430  PRINT "COORDINATES OF THE NODES"
1440  PRINT "===== == == ====="
1450  LPRINT CHR$(15)
1460  LPRINT "COORDINATES OF THE NODES"
1461  LPRINT "===== == == =====":LPRINT
1470  FOR I=1 TO NBE
1480      PRINT USING "##";I;
1490      PRINT USING " .####^####";X(I);Y(I)
1500      LPRINT USING "##";I;
1510      LPRINT USING " .####^####";X(I);Y(I)
1520  NEXT I

```

```

1521  LPRINT:LPRINT:LPRINT
1530  RETURN
1720  RETURN
1730  ' Boundary conditions
1740  '
1750  DIM COND(NBE,4),AIN(3)
1760  PRINT CHR$(12):FLAG=1
1770  PRINT CHR$(12):PRINT:PRINT:PRINT:PRINT
1780  PRINT "   You are required to input the boundary conditions."
1790  PRINT "   Mixed boundary conditions of Dirichlet and Neumann "
1800  PRINT "   type are allowed."
1810  PRINT "   Automatic generation using linear interpolation is"
1820  PRINT "   permitted":PRINT:PRINT
1821  PRINT "           REMEMBER !!!":PRINT
1822  PRINT "           NODE DATA           CODE"
1823  PRINT "           -----"
1824  PRINT "           q(bef), q(aft)      .... 1 "
1825  PRINT "           f , q(aft)         .... 2 "
1826  PRINT "           f , q(bef)         .... 3 "
1827  PRINT "           f , f (sharp)      .... 4 "
1828  PRINT "           f , f (smooth)     .... 5 "
1830  PRINT :PRINT :PRINT
1831  PRINT "if the data are going to be read from any file enter 'y'"
1840  PRINT "otherwise press any key to continue"
1850  AS=INKEY$:IF AS="" THEN GOTO 1850
1851  IF (AS<>"y" AND AS<>"Y") THEN GOTO 1850
1852  CLS:INPUT "file name";B$
1853  OPEN B$ FOR INPUT AS #1
1854  FOR NODO=1 TO NBE
1855      INPUT #1,COND(NODO,1),COND(NODO,2),COND(NODO,3),COND(NODO,4)
1856  NEXT NODO
1857  CLOSE #1:GOTO 2110
1860  PRINT CHR$(12)
1870  PRINT "Node number,Node code,f,δf/δn (back),δf/δn (forward)"
1880  PRINT "           Enter in that order if known"
1890  PRINT:PRINT :PRINT
1900  K=0
1901  JCOND4=0
1910  FOR I=1 TO NBE
1920      IF (K=NBE) THEN GOTO 2110
1930      INPUT "           ";L,COND(L,1),A1,A2
1940      '
1950      ' The information is interpreted according to the code
1960      '
1970      GOSUB 2540
1980      IF L=(K+1) THEN K=K+1:GOTO 2100
1990      INCR=L-K
2000      FOR IN=1 TO 3
2010          AIN(IN)=(COND(L,IN+1)-COND(K,IN+1))/INCR
2020      NEXT IN
2030      IN=INCR-1
2035      COND=COND(L,1)
2040      FOR M=1 TO IN

```

```

2044 KM=K+M
2045 COND(KM,1)=COND
2050 FOR MM=2 TO 4
2060 COND(K+M,MM)=COND(K,MM)+AIN(MM)*M
2070 NEXT MM
2080 NEXT M
2090 K=L
2100 NEXT I
2110 ' End of boundary conditions input
2120 CLS:PRINT " Do you want the boundary conditions to be printed ? (Y/N)"
2130 A$=INKEY$:IF A$="" THEN GOTO 2130
2135 IF (A$(">Y" AND A$(">y" ) THEN RETURN
2140 LPRINT CHR$(15)
2141 CLS:PRINT " Node Code      I      &I/&n (bef)  &I/&n (aft)"
2142 LPRINT "BOUNDARY CONDITIONS"
2143 LPRINT "=====I:LPRINT
2144 LPRINT " Node Code      I      &I/&n (bef)  &I/&n (aft)"
2145 LPRINT " -----"
2150 FOR I=1 TO NBE
2160 PRINT USING "### &I ";I;COND(I,1);
2170 PRINT USING " +.#####";COND(I,2);COND(I,3);COND(I,4)
2180 LPRINT USING "### &I ";I;COND(I,1);
2190 LPRINT USING " +.#####";COND(I,2);COND(I,3);COND(I,4)
2200 NEXT I
2202 RETURN
2210 ' The boundary is drawn
2220 '
2230 XMIN=X(1);XMAX=X(1);YMIN=Y(1);YMAX=Y(1)
2240 FOR I=1 TO NBE
2250 IF X(I)<XMIN THEN XMIN=X(I)
2260 IF Y(I)<YMIN THEN YMIN=Y(I)
2270 IF X(I)>XMAX THEN XMAX=X(I)
2280 IF Y(I)>YMAX THEN YMAX=Y(I)
2290 NEXT I
2300 DX=XMAX-XMIN
2310 DY=YMAX-YMIN
2320 EX=180/DX
2330 EE=180/DY
2340 IF (EX>ES) THEN ES=EX
2350 CLS
2360 SCREEN 1
2370 COLOR 1
2380 DEF FNxorN(XUS)=EE*(XUS-XMIN)+70
2390 DEF FNYORN(YUS)=190-EE*(YUS-YMIN)
2400 FOR I=1 TO NBE
2410 XI=FNxorN(X(I))
2420 YI=FNYORN(Y(I))
2430 IF (I=NBE) THEN XF=FNxorN(X(I));YF=FNYORN(Y(I));GOTO 2460
2440 XF=FNxorN(X(I+1))
2450 YF=FNYORN(Y(I+1))
2460 LINE (XI,YI)-(XF,YF),Z
2470 NEXT I

```

```

2471 FOR I=1 TO NBE
2472 XI=FNxorN(X(I))
2473 YI=FNYORN(Y(I))
2474 PSET(XI,YI),I
2475 NEXT I
2480 A$=INKEY$:IF A$="" THEN 2480
2481 'DEF SEG= &HBB00
2482 'BSAVE "bound",0,&H4000
2483 'DEF SEG
2490 SCREEN 2:SCREEN 0:COLOR 2
2500 RETURN
2510 '
2520 ' herein the boundary conditions are interpreted
2530 '
2540 CODE=COND(L,1)
2550 IF CODE=1 THEN COND(L,2)=0:COND(L,3)=A1:COND(L,4)=A2:RETURN
2560 IF CODE=2 THEN COND(L,2)=A1:COND(L,3)=0:COND(L,4)=A2:RETURN
2570 IF CODE=3 THEN COND(L,2)=A1:COND(L,3)=A2:COND(L,4)=0:RETURN
2580 IF CODE=4 THEN COND4=COND4+1:COND(L,2)=A1:COND(L,3)=0:COND(L,4)=0:RETURN
2590 IF CODE=5 THEN COND(L,2)=A1:COND(L,3)=0:COND(L,4)=0:RETURN
2600 RETURN
3000 '
3010 'herein q(aft) and q(bef) are calculated for type 4 nodes
3020 '
3030 FOR N=1 TO NBE
3040 IF COND(N,1)<>4 THEN GOTO 3250
3050 IF N=1 THEN NA=NBE ELSE NA=N-1
3060 IF N=NBE THEN ND=1 ELSE ND=N+1
3070 DFI=COND(N,2)-COND(NA,2)
3080 DYD=Y(ND)-Y(N)
3090 DFI=COND(ND,2)-COND(N,2)
3100 DYA=Y(N)-Y(NA)
3110 NUME=DFI*DYD-DXD*DYA
3120 DXA=X(N)-X(NA)
3130 DID=X(ND)-X(N)
3135 L1=SQR(DXA^2+DYA^2)
3136 L2=SQR(DID^2+DYD^2)
3140 DENO=DXA*DYD-DXD*DYA
3150 GRAY=NUME/DENO
3160 NUME=DXA*DFID-DFI*DXD
3170 GRAY=NUME/DENO
3210 NUME=GRAY*DYA-GRAY*DXA
3220 COND(N,3)=NUME/L1
3230 NUME=GRAY*DYD-GRAY*DXD
3240 COND(N,4)=NUME/L2
3250 NEXT N
3260 RETURN
9000 '
9010 ' This subroutine performs the assembly process
9020 ' Ind= node codes inside the element
9030 '
9040 CODE=COND(MNNN,1)
9041 NCO=NODO-ICOND4

```

```

9042 IF IND=1 THEN NF1=NF11 ELSE NF1=NF12
9050 IF IND=1 THEN B=B1:A=A1 ELSE B=B2:A=A2
9055 IF IND=1 THEN FLU=COND(NNNN,4) ELSE FLU=COND(NNNN,3)
9060 FUN=COND(NNNN,2)
9070 ON CODE GOTO 9090,9130,9170,9210,9230
9080 '
9090 COEF(NCO,NF1)=COEF(NCO,NF1)+A
9100 COEF(NCO,NEC+1)=COEF(NCO,NEC+1)+B*FLU
9110 RETURN
9120 '
9130 COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*FUN-(IND-2)*B*FLU
9140 COEF(NCO,NF1)=COEF(NCO,NF1)-B*(IND-1)
9150 RETURN
9160 '
9170 COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*FUN+(IND-1)*B*FLU
9180 COEF(NCO,NF1)=COEF(NCO,NF1)+B*(IND-2)
9190 RETURN
9200 '
9210 COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*FUN+B*FLU
9212 RETURN
9220 '
9230 COEF(NCO,NEC+1)=COEF(NCO,NEC+1)-A*FUN
9240 COEF(NCO,NF1)=COEF(NCO,NF1)-B
9250 RETURN
11000 ' Solution of the system of simultaneous equations
11010 '
11011 CLS:PRINT "The simultaneous linear equations are being solved"
11020 DIM VEC(NEC),AL(NEC+1)
11021 'FOR I=1 TO NEC
11022 '   FOR J=1 TO NEC+1
11023 '       PRINT COEF(I,J);
11024 '       LPRINT COEF(I,J);
11025 '   NEXT J
11026 'PRINT:LPRINT
11027 'NEXT I
11030 FOR I=1 TO NEC-1
11040   IN=0
11050   ME=0
11060   FOR J=I TO NEC
11070     IF (ABS(COEF(J,I))>ME) THEN ME=ABS(COEF(J,I)):IN=J
11080   NEXT J
11081   IF IN=0 THEN PRINT "singular matrix":STOP
11090   FOR J=1 TO NEC+1
11100     VI=COEF(I,J)
11110     COEF(I,J)=COEF(IN,J)
11120     COEF(IN,J)=VI
11130   NEXT J
11140   FOR K=I TO NEC+1
11150     AL(K)=COEF(I,K)/COEF(I,I)
11160   NEXT K

```

```

11170   FOR K=I+1 TO NEC
11180     MA=COEF(K,I)
11190     FOR J=I TO NEC+1
11200       COEF(K,J)=COEF(K,J)-AL(I)*MA
11210     NEXT J
11220   NEXT K
11230 NEXT I
11240 VEC(NEC)=COEF(NEC,NEC+1)/COEF(NEC,NEC)
11250 FOR K=NEC-1 TO 1 STEP -1
11260   SUMA=0
11270   FOR KK=K+1 TO NEC
11280     SUMA=SUMA-VEC(KK)*COEF(K,KK)
11290   NEXT KK
11300   VEC(K)=(COEF(K,NEC+1)+SUMA)/COEF(K,K)
11305 NEXT K
11310 RETURN
15000 '
15010 ' This subroutine interprets the results
15020 '
15025 ICOND4=0
15030 FOR NODD=1 TO NBE
15035   NNODD=NODD-ICOND4
15050   CODE=COND(NODD,1)
15060   ON CODE GOTO 15080,15090,15100,15110,15120
15070 '
15080   COND(NODD,2)=VEC(NNODD):GOTO 15125
15090   COND(NODD,3)=VEC(NNODD):GOTO 15125
15100   COND(NODD,4)=VEC(NNODD):GOTO 15125
15110   ICOND4=ICOND4+1:GOTO 15125
15120   COND(NODD,3)=VEC(NNODD):COND(NODD,4)=VEC(NNODD):GOTO 15125
15125 NEXT NODD
15130 RETURN
20000 '
20010 ' Solution printing
20020 '
20030 CLS:PRINT "The solution has been obtained, do you want it to be printed?"
20040 PRINT "":PRINT " (enter y or n)"
20050 A$=INKEY$:IF A$="" THEN GOTO 20050
20055 IF (A$<>"Y" AND A$<>"y") THEN RETURN
20056 LPRINT CHR$(15)
20060 CLS:PRINT " Node Code      δδ/δn (bef)  δδ/δn (aft)"
20061 LPRINT:LPRINT:LPRINT
20062 LPRINT "R E S U L T S "
20063 LPRINT "=====":LPRINT
20070 LPRINT " Node Code      δδ/δn (bef)  δδ/δn (aft)"
20071 LPRINT " ----"
20080 FOR I=1 TO NBE
20090   PRINT USING "###  ### ";I;COND(I,1);
20100   PRINT USING " +.#####^";COND(I,2);COND(I,3);COND(I,4)
20110   LPRINT USING "###  ### ";I;COND(I,1);
20120   LPRINT USING " +.#####^";COND(I,2);COND(I,3);COND(I,4)
20130 NEXT I
20140 RETURN

```